

Application of Cardinality based GRASP to the Biclustering of Gene Expression Data

Shyama Das
Department of Computer Science
Cochin University of Science and
Technology, Kochi, Kerala, India

Sumam Mary Idicula
Department of Computer Science,
Cochin University of Science and
Technology, Kochi, Kerala, India,

ABSTRACT

Biclustering algorithms perform simultaneous row and column clustering of a given data matrix. In gene expression dataset a bicluster is a subset of genes that exhibit similar expression patterns through a subset of conditions. Biclustering is a useful data mining technique for identifying local patterns from gene expression data. In this paper biclusters are identified in two steps. In the first step high quality bicluster seeds are generated using K-Means clustering algorithm. These seeds are then enlarged using Cardinality based Greedy Randomized Adaptive Search Procedure (CGRASP) which is a multi-start metaheuristic method in which there are two phases, construction and local search. The Experimental results on the benchmark datasets prove that CGRASP is capable of identifying biclusters of high quality compared to many of the already existing biclustering algorithms. Moreover far better biclusters are obtained in this algorithm compared to the already existing algorithm based on the same GRASP metaheuristics.

Categories and Subject Descriptors

H.2.8 [Information Systems]: Data Mining J.3 [Computer Applications]: Life and Medical Sciences

General Terms

Algorithms, Measurement, Experimentation,

Keywords

Gene expression data, greedy randomized adaptive search procedure, K-Means clustering, biclustering

1. INTRODUCTION

DNA Microarray technology revolutionized gene expression study by simultaneously measuring the expression levels of thousands of genes in a single experiment. Microarray Gene expression data is a high dimensional matrix where rows represent genes and columns represent experimental conditions or samples. The experimental conditions can be patients, tissue types, different time points etc. The gene expression datasets typically contain thousands of genes and hundreds of conditions. Each element in the matrix refers to the expression level of a particular gene under a specific condition. Each entry in this matrix is a real number. Genes participating in the same biological process will have similar expression patterns.

Clustering is the suitable mining method for identifying these patterns.

Being high dimensional mining various patterns from microarray gene expression data has immense application in bioinformatics and clinical research. These patterns are useful in the medical domain for aid in more accurate diagnosis, prognosis, treatment planning, drug discovery and protein network analysis. Clustering is the most widely used data mining technique for analyzing gene expression data to group similar genes or conditions. Clustering of co-expressed gene into biologically meaningful groups helps in inferring the biological role or function of new gene that is co-expressed with a known gene.

However clustering based on the entire row has many disadvantages and restrictions in implementation process. Clustering is based on the assumption that all the related genes behave similarly across all the measured conditions. It may reveal the genes which are very closely co-regulated. Based on a general understanding of the cellular process, the subsets of genes are co-regulated and co-expressed under certain experimental conditions. But they behave almost independently under other conditions. Moreover clustering happens to partition the genes into disjoint sets i.e. each gene is associated with a single biological function, which in fact is in contradiction to the biological system [1].

To overcome the problems of clustering concept of biclustering was introduced. Biclustering is clustering applied along the row and column, simultaneously. Clustering is a global model where as biclustering is a local model. This approach identifies the genes which show similar expression levels under a specific subset of experimental conditions. In biclustering the objective is to identify maximal subgroups of genes and subgroups of conditions such that the genes express highly correlated activities over a range of conditions. Biclustering was first introduced by Hartigan and called it direct clustering [2]. Cheng and Church were the first to apply biclustering to gene expression data [3]. Biclustering is also known as coclustering, bidimensional clustering and subspace clustering. The application of biclustering is ideal when some genes have multiple functions and experimental conditions are diverse.

In this work an algorithm is developed for biclustering gene expression data using CGRASP which is a semigreedy, multistart metaheuristics which alternates between construction and local search phase to find a globally optimal solution. Initially high quality bicluster seeds are generated using K-Means and they are enlarged using CGRASP.

2. BICLUSTERING

A bicluster is a submatrix of the gene expression data matrix. A bicluster of a gene expression dataset is a subset of genes which exhibit similar expression patterns along a subset of conditions. Let $X=\{G_1, G_2, \dots, G_N\}$ be the set of genes and $Y=\{C_1, C_2, \dots, C_M\}$ be the set of conditions in the gene expression dataset. The dataset can be viewed as an $N \times M$ matrix A of real numbers. A bicluster is a subset of rows that shows a coherent behaviour across a subset of columns and vice versa. A bicluster is a submatrix B of A and if the size of B is $I \times J$, then I is a subset of rows X of A , and J is a subset of the columns Y of A . The rows and columns of the bicluster B need not be contiguous as in the expression matrix A .

There are four types of biclusters namely biclusters with constant values, biclusters with constant values on rows or columns, biclusters with coherent values, and biclusters with coherent evolutions. Biclusters with coherent values are biologically more relevant than biclusters with constant values. In this work biclusters with coherent values are identified. Thus the problem of biclustering can be formulated as follows: given a data matrix A , find a set of submatrices B_1, B_2, \dots, B_n that satisfy some homogeneity characteristics or coherence. It is not necessary that the identified submatrices to be disjoint or to cover the entire matrix. A bicluster with coherent values identifies a subset of genes and a subset of conditions with coherent values on both rows and columns. The degree of coherence is measured by mean squared residue score or Hscore which was introduced by Cheng and Church. It is the sum of the squared residue score. The residue of an element reveals its degree of coherence with the other elements of the bicluster it belongs to. The residue score of an element b_{ij} in a submatrix B is defined as

$$RS(b_{ij}) = b_{ij} - b_{iJ} - b_{iI} + b_{IJ}$$

The residue score of an element b_{ij} provides the difference between the actual value and its expected value predicted from its row mean, column mean and bicluster mean. Hence from the value of residue, the quality of the bicluster can be evaluated by computing the mean squared residue. That is Hscore or mean squared residue score of bicluster B is

$$MSR(B) = \sum_{i \in I, j \in J} (RS(b_{ij}))^2 / (|I| * |J|)$$

Where

$$b_{iJ} = \sum_{j \in J} (b_{ij}) / (|J|)$$

$$b_{iI} = \sum_{i \in I} (b_{ij}) / (|I|)$$

$$b_{IJ} = \sum_{i \in I, j \in J} (b_{ij}) / (|I| * |J|)$$

Here I denotes the row set, J denotes the column set, b_{ij} denotes the element in a submatrix, b_{iJ} denotes the i th row mean, b_{iI} denotes the j th column mean, and b_{IJ} denotes the mean of the whole bicluster.

A bicluster B is called a δ bicluster if $MSR(B) < \delta$ for some $\delta > 0$. If the MSR value is high it means that the data is uncorrelated. If the MSR value is low then there is correlation in the matrix. The value of δ depends on the dataset and it should be calculated in advance. The value of δ for the two datasets are taken from [3]. These values are calculated from the clustering experiments done in [4]. For Yeast dataset the value of δ is 300 and for Lymphoma dataset the value of δ is 1200. The volume of a bicluster or

bicluster size is the product of number of rows and the number of columns in the bicluster. Larger the volume and smaller the MSR or Hscore of the bicluster better is the quality of the bicluster

Biclustering problem is computationally intractable. Computation of biclusters is costly because one will have to consider all the combinations of columns and rows in order to find out all the biclusters. The search space for the biclustering problem is very large with size 2^{m+n} where m and n are the number of rows and columns of the gene expression data matrix respectively. Usually $m+n$ is more than 2000. The biclustering problem is Np-hard.

In this algorithm each bicluster is represented by a binary string of fixed length $n+m$, where n and m are the number of genes and conditions of the microarray dataset, respectively. The first n bits are associated to n genes, the following m bits to m conditions. If a bit is set to 1, it means that the corresponding gene or condition belongs to the bicluster; otherwise it does not. This encoding presents the advantage of having fixed size [5].

3. ALGORITHM DESCRIPTION

Biclustering problem can be solved using different algorithm design techniques such as iterative row and column clustering combination, Divide and Conquer, Greedy iterative search, Evolutionary or metaheuristic algorithms. In this work the multi-start metaheuristic method called Cardinality based Greedy Randomized Adaptive Search Procedure (CGRASP) is used for finding δ biclusters. The algorithm has two major phases. In the first phase, an initial set of seed biclusters are generated using K-Means one way clustering algorithm. The second phase is used to enlarge the seeds by adding more rows and columns using CGRASP. Greedy seed growing strategy makes a choice that optimizes a local gain in the hope that this choice will lead to a globally good solution but has local minima problem. Metaheuristic methods have the potential to escape from local minima. Moreover GRASP is semi-greedy. Hence it can combine the advantages of both greedy and random solution constructions.

3.1. Seed Finding

A seed is a tightly co-regulated small bicluster with a possibility of accommodating more genes and conditions within the given MSR threshold. Seeds are identified as follows. Using the K-Means clustering algorithm the gene expression dataset is partitioned into n gene clusters and m sample clusters. In order to get maximum 10 genes per gene cluster, it is further divided according to the cosine angle distance from the cluster centre. Similarly each sample cluster is further divided into sets of 5 samples according to cosine angle distance from the cluster centre. The number of gene clusters having maximum 10 close genes is p and the number of sample clusters having maximum 5 conditions is q . The initial gene expression data matrix is thus partitioned into $p \times q$ submatrices. The MSR value of the submatrices is calculated and those with MSR value below a certain limit are selected as seeds [6]. The Yeast dataset is partitioned into 140 gene clusters and 3 condition clusters. The Lymphoma dataset is partitioned into 200 gene clusters and 15 condition clusters [5].

3.2. Seed Growing using Cardinality based Greedy Randomized Adaptive Search Procedure (CGRASP)

GRASP is a multi-start metaheuristics for solving combinatorial Optimization problems. GRASP is an iterative randomized sampling method in which each iteration consists of two phases: construction and local search. The construction phase generates a feasible solution, whose neighborhood is investigated until a local minimum is identified during the process of local search phase. The best overall solution is reserved as the result. In the construction phase a feasible solution is iteratively developed by adding one element at a time. During each iteration of the construction phase a set of candidate elements are formed by all the elements that can be incorporated to the partial solution under construction without eliminating feasibility. The selection of the next element for incorporation is resolved by the evaluation of all candidate elements in accordance with a greedy evaluation function.

This greedy function stands for the incremental increase in the cost function because of the incorporation of this element into the solution under construction. The evaluation of the elements by this function results in the creation of a restricted candidate elements (RCL) produced by the best elements. That is, those elements whose incorporation to the current partial solution results in the smallest incremental costs. This is the greedy aspect of the algorithm. The element which is to be incorporated into the partial solution is randomly chosen from those in the RCL. This is the probabilistic aspect of the heuristic algorithm. Once the chosen element is included in the partial solution, the candidate list is restructured and the incremental costs are recalculated. This is the adaptive aspect of the heuristic algorithm. The restricted candidate list RCL is constituted of elements with the best (i.e., the smallest) incremental costs. This list can be controlled by different factors. That is, either by the number of elements (cardinality-based) or by their quality (value-based). In the first case it is constituted of the p elements with the best incremental costs where p is a parameter [7].

The solutions produced by the greedy randomized construction are not always optimal even with respect to simple neighborhoods. The local search phase makes the constructed solution better. A local search algorithm functions in an iterative manner by consecutively replacing the current solution by an enhanced solution in the neighborhood of the existing solution. It finishes when no better solution is identified in the neighborhood. There are two ways to implement local search: using first improving strategy or best improving strategy. For the best improving strategy the current solution is replaced by the best neighbor through an investigation of all the neighbors. On the other hand in the case of the first improving strategy the current solution is always replaced by the first neighbor whose cost function value is smaller than that of the current solution [8].

While applying GRASP to gene expression data, conditions and genes are added to the seed in the construction phase. For this purpose the list of genes or conditions not included in the bicluster are identified. From this list, candidate list is formed by those genes or conditions whose inclusion in the bicluster will not

exceed the Hscore or MSR value above the given MSR threshold. The candidate lists is dynamic in the sense that it varies as the bicluster size varies. From the candidate lists the best elements are selected and another list is formed known as restricted controlled list or RCL. RCL contains genes or conditions which when added results in incremental Hscore or MSR value less than a threshold known as RCL threshold. RCL threshold is calculated using the formula $MSR_{min} + \alpha (MSR_{max} - MSR_{min})$. Here MSR_{max} is the maximum MSR value obtained when a single gene or condition is added from the candidate list. MSR_{min} is the minimum value of MSR when a gene or condition is added from the candidate list for a given iteration. The value of α ranges from 0 to 1. The parameter α controls the amounts of greediness and randomness. The case $\alpha=0$ corresponds to pure greedy algorithm, where as $\alpha=1$ is equivalent to a random construction in the algorithm. For cardinality based grasp p best elements are selected.

From the RCL an element is selected at random and added to the bicluster. After adding an element from the RCL the candidate list has to be updated and the process is continued. This is the construction phase. After the construction phase local search is performed. Thus construction and local search is continued alternately till the mean square residue score of the bicluster reaches the given threshold. Here the neighborhood search is implemented using first-improving strategy. To get biclusters having more conditions the gene list and condition list are maintained separately. Thereafter the construction phase is executed first from the condition list and then from the gene list.

Procedure construct_candidatelist (bicluster, δ)

```

Bicluster1 ← bicluster;

notinlist ← the list of Genes or Conditions not
included in the bicluster

notinlistcount ← noofelements(notinlist)

For i=1:notinlistcount
    Hscorelist[i] = Hscore(Bicluster1 U
notinlist[i])
End(for)

Candidatelist = {}

For i=1:notinlistcount
    If Hscorelist[i] <  $\delta$ 
        Candidatelist = Candidatelist U Notinlist[i]
    End(for)
end(construct_candidatelist)

```

Procedure Greedy_Randomized_Construct (Seed)

```

bicluster←Seed;
p is assigned the cardinality value
While solution construction notdone
    cand←construct_candidateList (bicluster, δ)
    RCL←BuildRCLcardinalitybased(bicluster,
        cand,p)
    Select an element S from RCL at random
    bicluster=bicluster U {S}
    Update G or C
End(while)
End(Greedy_Randomized_Construct)

```

Procedure

BuildRCLcardinalitybased(bicluster,C,P)

```

// C is the candidate list
//P is the cardinality
Sminhscore = inf
Smaxhscore = -inf
nocan=noofelements(C)
for I=1:nocan do
    calculate H[i]← Hscore{ bicluster U C[i]}
    if H[i ]<Sminhscore
        Sminhscore=H[i]
    Endif
    if H[i ]>Smaxhscore
        Smaxhscore=H[i]
    Endif
Endfor
RCLthresh=Sminhscore+α*(Sminhscore-
    Sminhscore)
RCL1={ }
For i=1:nocan
    If H[i]<RCLthresh
        RCL1=RCL1 U { C[i]}
    endif

```

```

end(for)
RCL= P best elements from RCL1
end (BuildRCLcardinalitybased)

```

Procedure Local_Search(bicluster)

```

// uses first-improving strategy
While there exists S e genelist or conditionlist do
    If Hscore(bicluster U S)<Hscore(bicluster)
        bicluster={ bicluster U S}
        break;
    endif
end(while)
end(Local_Search)

```

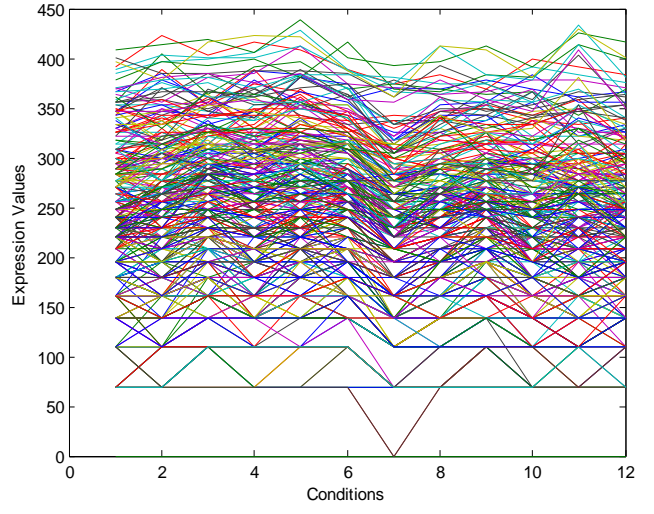
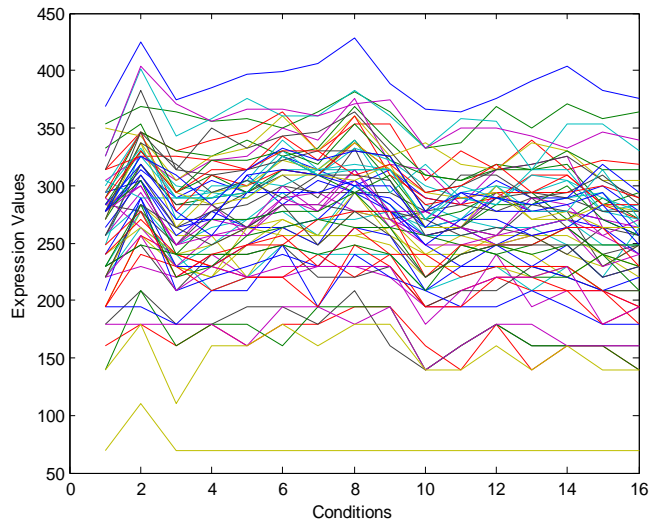
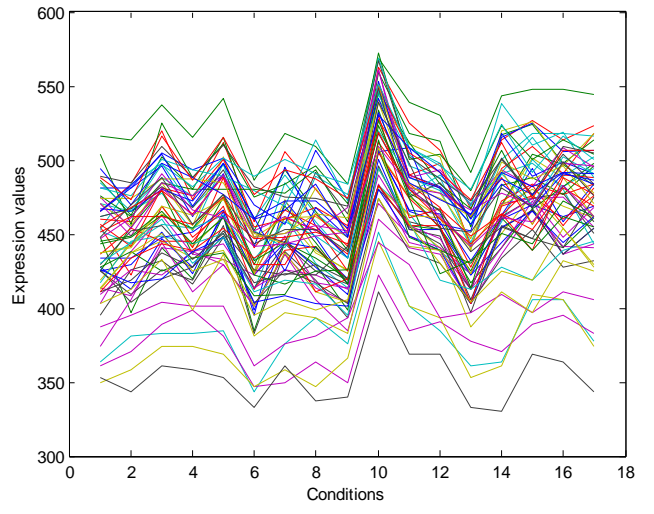
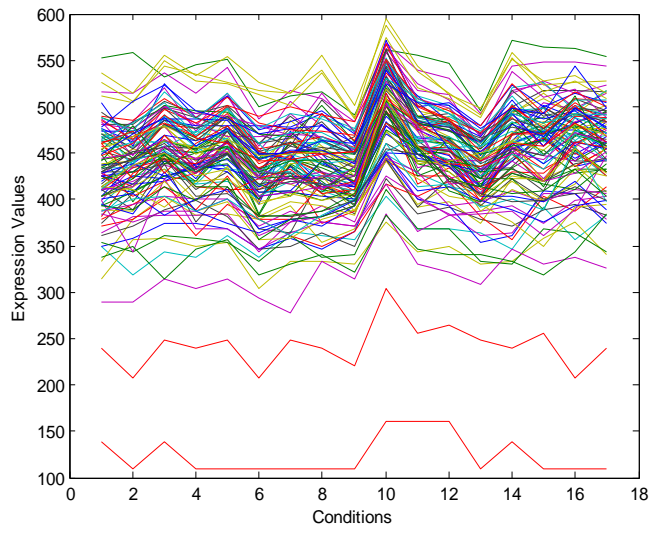
4. EXPERIMENTAL RESULTS

4.1. Datasets used

Two benchmark datasets namely Yeast *Saccharomyces Cerevisiae* Cell Cycle expression dataset and Lymphoma dataset are used in order to conduct the experiments and thus evaluate the quality of the proposed algorithm. The algorithm is implemented in Matlab. The Yeast dataset originated from [9]. Yeast dataset consists of 2884 genes and 17 conditions. The expression values were transformed by scaling and logarithm $x \rightarrow 100 \log(10^3 x)$. The values in the expression dataset after this transformation are integers in the range 0 to 600. Missing values are represented by -1. Human B-cell Lymphoma expression data contain 4026 genes and 96 conditions. The dataset was downloaded from the website for supplementary information for the article by Alizadeh et al. (2000) [10]. The expression levels were reported as log ratios. After scaling by a factor of 100 the values in the dataset are integers in the range -750 to 650. There are 47,639 (12.3%) missing values in the Lymphoma dataset. Missing values were represented by 999. The datasets after the above preprocessing is obtained from <http://arep.med.harvard.edu/biclustering>. In the Lymphoma dataset missing values are replaced by random numbers between -800 and 800 as in ref [3].

4.2. Bicluster Plots for Yeast dataset

In Figure 1 eight biclusters obtained using CGRASP is shown. Biclusters with all 17 conditions is obtained using this method. In the already existing works based on the same GRASP metaheuristics there is no bicluster with all 17 conditions. From the bicluster plots which show strikingly similar upregulation and down regulation we can conclude that CGRASP is an ideal method for identifying highly coherent biclusters from gene expression data.



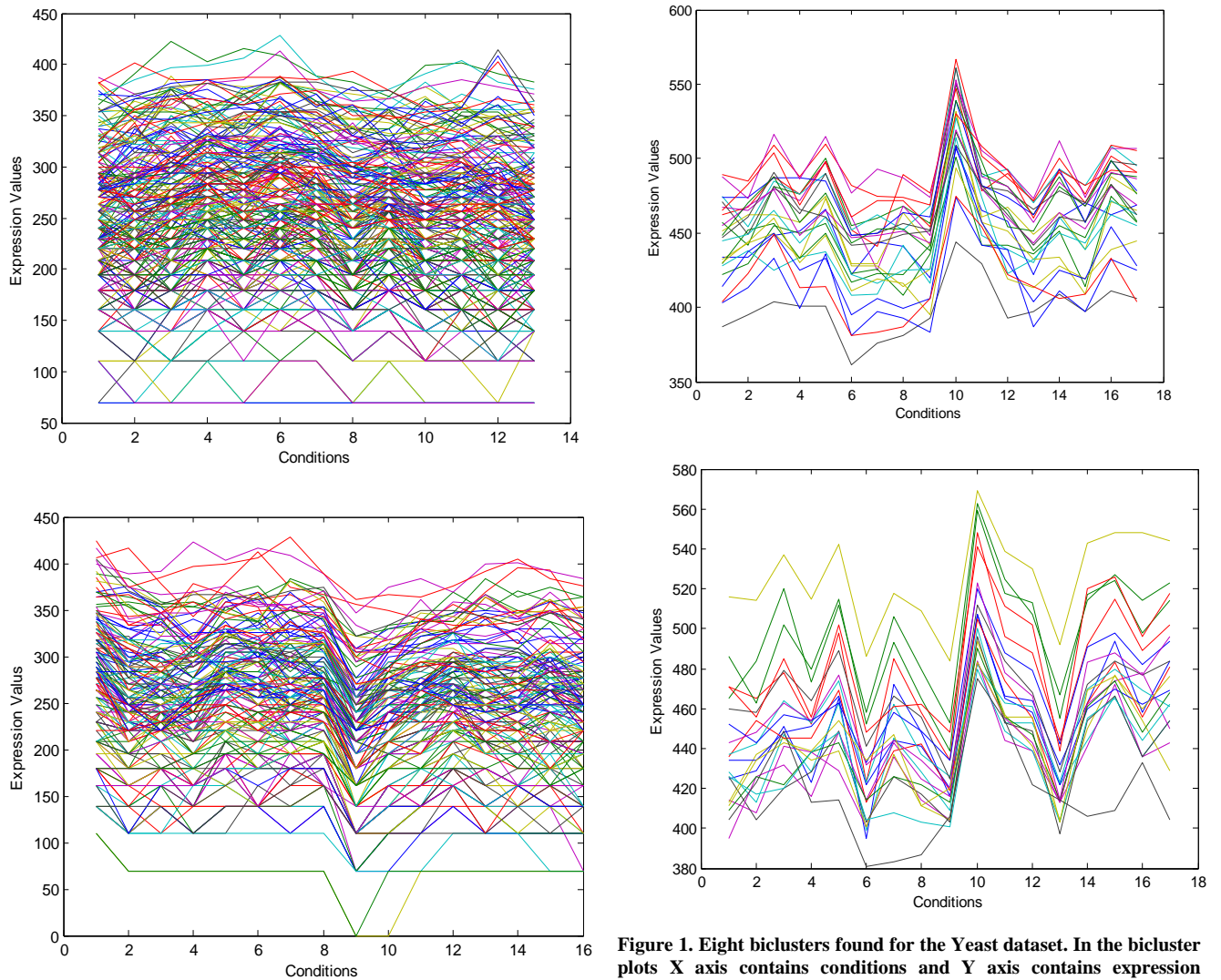


Figure 1. Eight biclusters found for the Yeast dataset. In the bicluster plots X axis contains conditions and Y axis contains expression values.

Bicluster labels are (a), (b), (c), (d), (e) and (f) respectively The details about biclusters can be obtained from Table 1 using bicluster label. All the means squared residues are lower than 200.

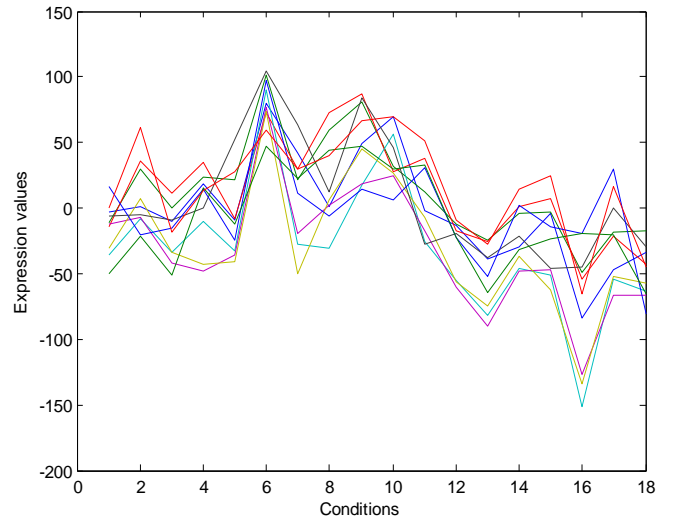
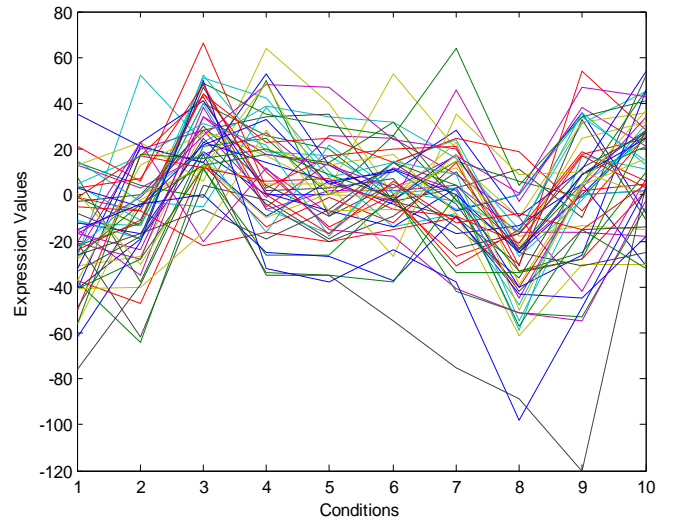
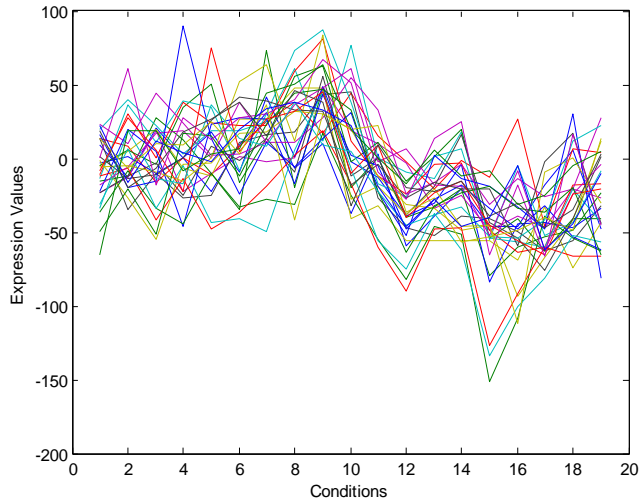
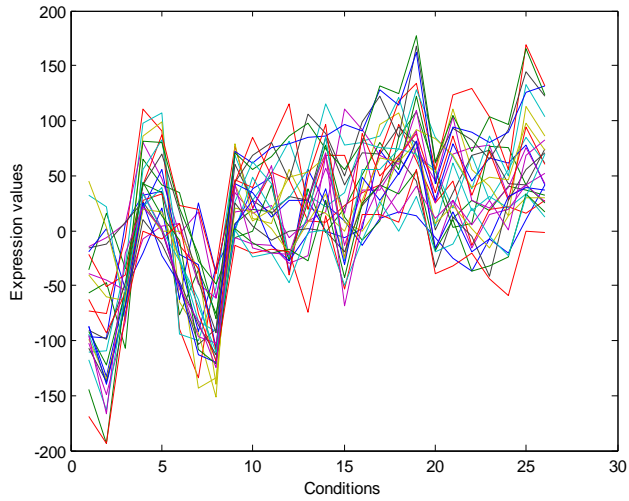
Table 1. Information about Biclusters of Figure 1.

Bicluster label	Rows	Columns	Bicluster Volume	MSR
(a)	107	17	1819	199.1857
(b)	64	16	1024	149.6244
(c)	63	17	1071	148.1866
(d)	324	12	3888	193.7751
(e)	256	13	3328	199.7194
(f)	164	16	2624	199.7293
(g)	24	17	408	87.2199
(h)	21	17	357	94.4589

In the above table the first column contains the label of each bicluster. The second and third columns report the number of rows (genes) and of columns (conditions) of the bicluster respectively. The fourth column reports the volume of the bicluster and the last column contains the mean squared residue or Hscore of the bicluster.

4.3. Bicluster Plots for Lymphoma dataset

GRASP metaheuristics are not applied to the biclustering of Lymphoma dataset in any of the previously published literature. Eight biclusters obtained by applying CGRASP to Lymphoma dataset are shown in Figure 2.



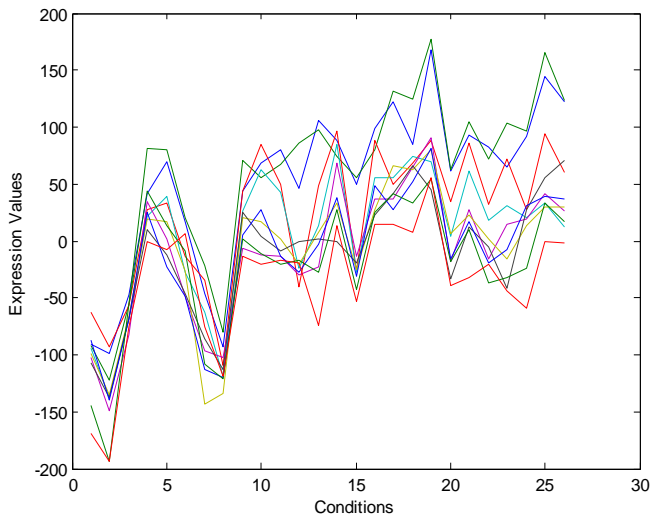
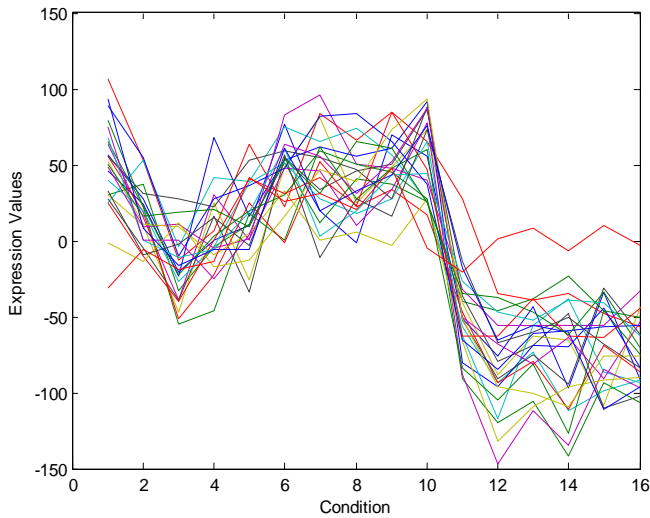
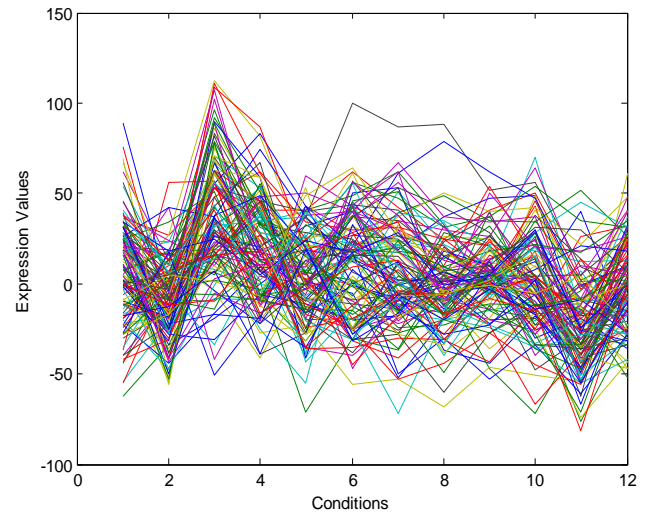
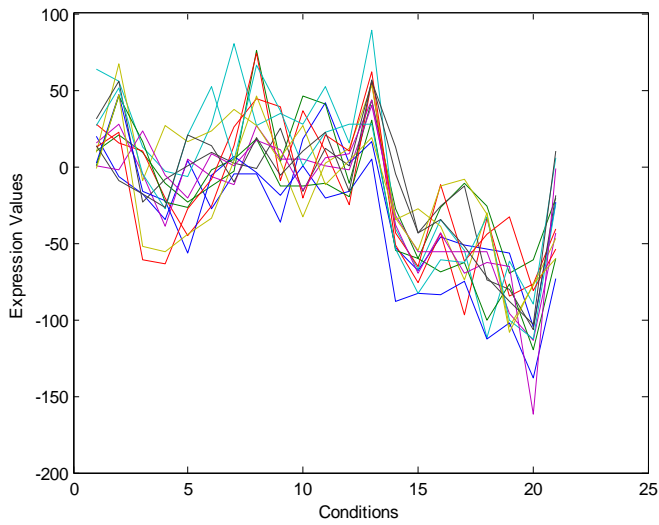


Figure 2. Eight biclusters found for the Lymphoma dataset. In the bicluster plots X axis contains conditions and Y axis contains expression values.

The bicluster labels are (p), (q), (r), (s), (t), (u), (v) and (w) respectively. The details of the biclusters are given in Table 2.

Table 2. Information about Biclusters of Figure 2

Label	Rows	Columns	Volume	MSR
(p)	26	26	676	883.6869
(q)	30	19	570	441.5052
(r)	52	10	520	307.5545
(s)	10	18	180	368.0541
(t)	14	21	294	409.6572
(u)	24	16	384	542.8357
(v)	10	26	260	388.4876
(w)	112	12	1344	492.4187

In the table given above the first column contains the label of each bicluster. The second and third columns report the number of rows (genes) and of columns (conditions) of the bicluster respectively. The fourth column reports the volume of the bicluster and the last column contains the mean squared residue or Hscore of the bicluster.

5. COMPARISON

The Table 3 given below provides a summary of results obtained by related algorithms on Yeast dataset. All the algorithms listed in Table 3 are having MSR value more or less equal to 200, even though the maximum limit of δ is 300. The performance of CGRASP algorithm in comparison with that of value based GRASP [11], CGRASP [12], RGRASP [13], SEBI [14], Cheng and Church's algorithm (CC) [3], and the algorithm FLOC by Yang et al. [15] and DBF [16] etc are given. SEBI (Sequential Evolutionary Biclustering) used evolutionary computation for solving biclustering problem. CC algorithm used greedy approach. The CC algorithm starts with the entire gene expression data matrix and removed rows and columns from the gene expression data matrix to find a bicluster. The model of bicluster proposed by Cheng and Church was generalized by Yang et al (2003) for incorporating null values and for removing random interference. They developed FLOC which is a probabilistic algorithm that can discover a set of possibly overlapping biclusters simultaneously. Deterministic Biclustering with frequent pattern mining (DBF) was proposed by Zhang et al. DBF generates good quality bicluster seeds using frequent pattern mining. In the second phase these seeds are enlarged by adding more genes or conditions. In the case of CGRASP algorithm presented here average number of genes is better than that of all other algorithms except CC, FLOC and DBF. Average number of conditions is better than that of all other algorithms. Average volume is better than that of all other algorithms. Average MSR value is lower than that of all other algorithms except DBF. Largest bicluster size is better than that of all other algorithms except CC and DBF. The biclusters obtained using CGRASP is better than that of GRASP in [11], CGRASP [12] and RGASP [13] in terms of average gene number, average condition number, average volume, average MSR and largest bicluster size. In this method there are biclusters with all 17 conditions for Yeast datasets. In the already existing works based on GRASP there is no bicluster with all 17 conditions. In metaheuristic methods like multi-objective evolutionary computation [17] the maximum number of conditions obtained is only 11 for Yeast dataset.

Table 3. Performance comparison between CGRASP and other algorithms for Yeast dataset

Algori thm	Avg. gene num	Avg. cond. num.	Avg. Volume	Avg. MSR	Largest Bicluster size
CGRASP in this study	163.00	15.17	2292.33	181.70	3888
GRASP [11]	30.00	14.00	430.33	188.57	1335
CGRASP [12]	18.20	12.20	215.40	187.05	319
RGRASP [13]	21.25	13.13	283.38	182.34	854
SEBI	13.61	15.25	209.92	205.18	1394
CC	166.71	12.09	1576.98	204.29	4485
FLOC	195.00	12.80	1825.78	187.54	2000
DBF	188.00	11.00	1627.20	114.70	4000

In the above table the average mean squared residue, the average number of genes and conditions, average volume and largest bicluster size are compared for various algorithms. For the average mean squared residue field lower values are better where as higher values are better for all other fields.

For Lymphoma dataset the value of average MSR is very low compared with SEBI and CC. Hence naturally the average gene number, average number of conditions and average volume will be less. But in spite of that the average number of genes obtained is much greater than that of SEBI.

Table 4. Performance comparison between CGRASP and other algorithms for Human Lymphoma dataset

Algori thm	Avg.gene. Num.	Avg. cond.num	Avg. Volume	Avg. MSR
CGRASP	34.75	18.50	528.50	479.27
SEBI	14.07	43.57	615.84	1028.84
CC	269.22	24.50	4595.98	850.04

As clear from the above table the average mean squared residue, the average number of genes and conditions and average volume are compared for various algorithms. For the average mean squared residue field lower values are better where as higher values are better for all other fields.

6. CONCLUSION

In this paper a new algorithm is introduced based on the CGRASP metaheuristics method for finding biclusters in gene expression data. In the first step K-Means algorithm is used to group rows and columns of the data matrix separately. Then they are combined to produce submatrices. From these submatrices those with Hscore value below a certain threshold are selected as seeds which are small tightly co-regulated submatrices. Then more genes and conditions are added to these seeds using CGRASP. This algorithm is implemented on the Yeast *Sacharomyces cerevisiae* cell cycle expression dataset and also the Human Lymphoma dataset. The results for Yeast dataset indicate that CGRASP algorithm performed better than other approaches. Biclusters discovered are larger having more genes and conditions with low Hscore value.

In short CGRASP method finds high quality biclusters that show strikingly similar up-regulations and down-regulations under a set of experimental conditions that can be inspected visually by using plots. The quality of biclusters found by CGRASP approach in this study is better than already existing biclustering algorithms. Moreover far better biclusters are obtained in this algorithm compared to the already existing algorithms based on the same GRASP metaheuristics. This is the first time that GRASP metaheuristics and its variants are applied in finding biclusters of Human Lymphoma dataset.

7. REFERENCES

- [1] Madeira S. C. and Oliveira A. L., “Biclustering algorithms for Biological Data analysis: a survey” IEEE Transactions on computational biology and bioinformatics, 2004. pp. 24-45.
- [2] J. A. Hartigan, “Direct clustering of Data Matrix”, Journal of the American Statistical Association, 1972. Vol.67, no.337, pp. 123-129.
- [3] Yizong Cheng and George M. Church, “Biclustering of expression data”, Proc. 8th Int. Conf., Intelligent Systems for Molecular Biology, 2000. pp. 93-103.
- [4] Tavazoie S., Hughes J. D., Campbell M. J., Cho R. J. and Church G. M., “Systematic determination of genetic network architecture”, Nat. Genet., 1999. Vol. 22, no.3 pp. 281-285.
- [5] Anupam Chakraborty and Hitashyam Maka “Biclustering of Gene Expression Data Using GeneticAlgorithm” Proceedings of Computation Intelligence in Bioinformatics and Computational Biology, CIBCB, 2005. pp. 1-8.
- [6] Chakraborty A. and Maka H., “Biclustering of gene expression data by simulated annealing”, HPCASIA 2005, pp. 627-632.
- [7] Feo TA and Resende MGC “Greedy Randomized Adaptive Search Procedures”, Journal of Global Optimization Vol. 6 1995. pp. 109-133.
- [8] Mauricio G.C Resende and Celso C, Rebeiro, “Greedy Randomized Adaptive Search Procedures” in *Handbook of Metaheuristics*, Edited by Fred Glover and Gary A Kochenberger, Kluwer Academic Publishers, New York, 2003. p. 221.
- [9] R.Cho, M.Campbell, E.Winzeler, L.Steinmetz, A.Conway, L.Wodika, T. Wolfsberg, A. Gabrielian, D. Landsman, D.Lockhart and R.Davis, “A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle,” Molecular Cell, 1998. Vol. 2, pp. 65-73.
- [10] Alizadeh, A. A. et al., “Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling”, Nature, 2000. Vol.43, no. 6769, pp. 503-11.
- [11] Dharan S and Nair ASN, “Biclustering of gene expression data using greedy randomized adaptive search procedure”, IEEE TENCON 2008. pp. 1-5.
- [12] Smitha Dharan and Achuthsankar S Nair, “Cardinality based Greedy Randomized Adaptive Search Algorithm for the detection of biclusters in Microarray gene expression data”, Proc. Int. Conf. Advanced Computing and Communication Technologies for High Performance Applications, 2008, Vol. 1, pp. 244-248.
- [13] Smitha Dharan, Achuthsankar S. Nair, "Biclustering of Gene expression Data using Reactive Greedy Randomized Adaptive Search Procedure", BMC Bioinformatics, 2009. Vol. 10, Suppl 1: s27.
- [14] Federico Divina and Jesus S. Aguilar-Ruize, “Biclustering of Expression Data with Evolutionary computation”, IEEE Transactions on Knowledge and Data Engineering, 2006. Vol. 18, pp. 590-602.
- [15] J. Yang, H. Wang, W. Wang and P. Yu, “Enhanced Biclustering on Expression Data”, Proc. Third IEEE Symp. Bioinformatics and BioEng. (BIBE'03), 2003. pp. 321-327.
- [16] Z. Zhang, A. Teo, B. C. Ooi, K. L. Tan, “Mining deterministic biclusters in gene expression data”, In: Proceedings of the fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE'04), 2004, pp. 283-292.
- [17] Banka H. and Mitra S., “Multi-objective Evolutionary biclustering of gene expression data”, Journal of Pattern Recognition, 2006. Vol.39, pp. 2464-2477.